DETAILED ACTION

1. Claims 1-35 are pending.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 23-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

4. Claims 23-27 refer to a "program". Programs are non-statutory subject matter, because signals do not fall within one of the four statutory categories of patentable subject matter: "process", "machine", "manufacture" or "composition of matter".

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 7,373,496 to Sekiguchi et al., in view of U.S. Reissue Patent 40,092 to Kang.

7. As to claim 1, Sekiguchi discloses a rapid restart method comprising: saving, before restart of an OS, process information in the OS relating to a user process [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5] to a save area on a main memory device [500]; initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process [col. 10, lines 6-10]; and restoring the saved process information in the OS after the restart of the OS [col. 10, line 29 – col. 11, line 35].

Sekiguchi teaches the limitations of the claim, but does not specifically teach that the process information comprises a process condition.

Kang teaches storing information during a restart of a computer [col. 2, lines 65-67, and col. 3, lines 1-15]. Thus, Kang teaches an information storage means similar to that of Sekiguchi. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

At the time that the invention was made, it would have been obvious to a person of ordinary skill in the art to employ the status storing means as taught by Kang. One of ordinary skill in the art would have been motivated to do so that the status of the computer can be stored during a restart operation.

It would have been obvious to one of ordinary skill in the art to combine the teachings of the cited references because they are both directed to the problem of storing information during a restart of a computer. Moreover, the status storing means taught by Kang would improve the flexibility of Sekiguchi because it allowed process conditions to be stored and restored after a restart..

8. As to claim 2, Sekiguchi discloses a rapid restart method comprising: saving, before restart of an OS, process information in the OS relating to a user process to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5], while setting a restart flag for the saved process information to designate whether the process is to be restarted or not [col. 10, lines 14-20]; initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process for which the restart flag is set not to restart [col. 10, lines 6-10]; and restoring the saved process information of the user process for which the restart flag is set not to restart in the OS, after the restart of the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

9. As to claim 3, Sekiguchi discloses a rapid restart method comprising: saving, before restart of an OS, process information in the OS relating to a user process to be continuously operated after restart of the OS, to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process [col. 10, lines 6-10]; and restoring the saved process information in the OS, after the restart of the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

10. As to claim 4, Sekiguchi discloses a rapid restart method comprising: saving, before restart of an OS, process information in the OS relating to a user process to be continuously operated after restart of the OS, to a save area on a main memory device by referring to a process ID table in which an identifier for a process to be continuously operated or a process not to be continuously operated after restart of the OS [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process [col. 10, lines 6-10]; and restoring the saved process information in the OS, after the restart of the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

11. As to claim 5, Sekiguchi discloses a rapid restart method comprising: saving, at generation of a user process, process information in the OS relating to the generated user process to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; setting, at switching of the user process, a restart flag for the saved process information to designate whether the process is to be restarted or not, while updating the process information saved in the save area to the latest state if the process is not to be restarted [603]; nullifying the saved process information, at termination of the user process [1004]; initializing, at restart of an OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process for which the restart flag is set not to restart [col. 10, lines 6-

10]; and restoring, after the restart of the OS, the saved process information of the user process for which the restart flag is set not to restart in the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

12. As to claim 6, Sekiguchi discloses when a restart flag is set for process information relating to a certain user process to designate whether the process is to be restarted or not, all the user processes belonging to the same user application program as the user process are searched, and restart flags in the process information relating to all the searched user processes are also set to the same value [col. 8, lines 19-37].

13. As to claim 7, Sekiguchi discloses when a restart flag is set for process information relating to a certain user process to designate whether the process is to be restarted or not, all the user processes belonging to the user application program as the user process are searched, and restart flags in the process information relating to all the searched user processes are also set to the same value [col. 8, lines 19-37].

14. As to claim 8, Sekiguchi discloses the OS is started up from an OS main memory image stored in a nonvolatile storage portion forming a part of the main memory device [FIG. 1, "REBOOT LOADER" in main memory device 102].

15. As to claim 9, Sekiguchi discloses every time occurs a write access from the OS to the nonvolatile storage portion during system operation, data in an address range having a predetermined width including the address at which the write access has occurred is copied from the nonvolatile storage portion to a substitute area ensured in a readable/writable main memory portion forming a part of the main memory device, and subsequent accesses to the address range having the predetermined width are converted to accesses to the substitute area [col. 9, line 39—col. 10, line 5].

16. As to claim 10, Sekiguchi discloses an information processing apparatus comprising: process information saving means for saving, before restart of an OS, process information relating to a user process to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; main memory initialization means for initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process [col. 10, lines 6-10]; and process restoration means for restoring the saved process information in the OS, after the restart of the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

17. As to claim 11, Sekiguchi discloses an information processing apparatus comprising: process information saving means for saving, before restart of an OS, process information relating to a user process to a save area on a main memory device [col. 13, lines 14-52, and col.

9, line 39—col. 10, line 5]; restart flag setting means for setting a restart flag for the saved process information to designate whether the process is to be restarted or not [col. 10, lines 14-20]; main memory initialization means for initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process for which the restart flag is set not to restart [col. 10, lines 6-10]; and process restoration means for restoring, after the restart of the OS, the saved process information of the user process for which the restart flag is set not to restart in the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

18. As to claim 12, Sekiguchi discloses an information processing apparatus comprising: process information saving means for saving, before restart of an OS, process information in the OS relating to a user process to be continuously operated after restart of the OS, to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; main memory initialization means for initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process [col. 10, lines 6-10]; and process restoration means for restoring the saved process information in the OS, after the restart of the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

19. As to claim 13, Sekiguchi discloses before the restart of the OS, the process information saving means saves process information in the OS relating to a user process to be continuously operated after restart of the OS, to the save area on the main memory device, by referring to a process ID table storing identifiers of processes to be continuously operated or of processes not to be continuously operated [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5].

20. As to claim 14, Sekiguchi discloses an information processing apparatus comprising: process save area generating means for saving, at generation of a user process, process information in the OS relating to the generated user process to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; process save information updating means for setting, at switching of the user process, a restart flag for the saved process information to designate whether the process is to be restarted or not, while updating the process information saved in the save area to the latest state if the process is not to be restarted [col. 10, lines 14-20]; process information save area releasing means for nullifying the saved process information, at termination of the user process [1004]; main memory initialization means for initializing, at restart of an OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process for which the restart flag is set not to restart [col. 10, lines 6-10]; and process restoration means for restoring, after the restart of the OS, the saved process information of the user process for which the restart flag is set not to restart, in the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

21. As to claim 15, Sekiguchi discloses means for searching, when a restart flag is set for process information relating to a certain user process to designate whether the process is to be restarted or not, all the user processes belonging to the same user application program as the user process, and setting restart flags in the process information relating to all the searched user processes to the same value [col. 8, lines 19-37].

22. As to claim 16, Sekiguchi discloses means for searching, when a restart flag is set for process information relating to a certain user process to designate whether the process is to be restarted or not, all the user processes belonging to the user application program as the user process, and setting restart flags in the process information relating to all the searched user processes to the same value [col. 8, lines 19-37].

23. As to claim 17, Sekiguchi discloses means for starting up the OS from an OS main memory image stored in a nonvolatile storage portion forming a part of the main memory device [FIG. 1, "REBOOT LOADER" in main memory device 102].

24. As to claim 18, Sekiguchi discloses means for copying, at every occurrence of a write access from the OS to the nonvolatile storage portion during system operation, data in an address range having a predetermined width including the address at which the write access has occurred from the nonvolatile storage portion to a substitute area ensured in a readable/writable main memory portion forming a part of the main memory device, and for converting subsequent

accesses to the address range having the predetermined width to accesses to the substitute area [col. 9, line 39—col. 10, line 5].

25. As to claim 19, Sekiguchi discloses a program for causing a computer to function as: process information saving means for saving, before restart of an OS, process information in the OS relating to a user process to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; main memory initialization means for initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process [col. 10, lines 6-10]; and process restoration means for restoring the saved process information in the OS after the restart of the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

26. As to claim 20, Sekiguchi discloses a program for causing a computer to function as: process information saving means for saving, before restart of an OS, process information relating to a user process to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; restart flag setting means for setting a restart flag for the saved process information to designate whether the process is to be restarted or not [col. 10, lines 14-20]; main memory initialization means for initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process for which the restart flag is set not to restart [col. 10, lines 6-10]; and process

restoration means for restoring, after the restart of the OS, the saved process information of the user process for which the restart flag is set not to restart in the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

27. As to claim 21, Sekiguchi discloses a program for causing a computer to function as: process information saving means for saving, before restart of an OS, process information in the OS relating to a user process to be continuously operated after restart of the OS, to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; main memory initialization means for initializing, at the restart of the OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process [col. 10, lines 6-10]; and process restoration means for restoring, after the restart of the OS, the saved process information in the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

28. As to claim 22, Sekiguchi discloses before the restart of the OS, the process information saving means saves process information in the OS relating to a user process to be continuously operated after restart of the OS, to the save area on the main memory device, by referring to a process ID table storing identifiers of processes to be continuously operated or of processes not to be continuously operated [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5].

29. As to claim 23, Sekiguchi discloses a program for causing a computer to function as: process save area generating means for saving, at generation of a user process, process information in the OS relating to the generated user process to a save area on a main memory device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; process save information updating means for setting, at switching of the user process, a restart flag for the saved process information to designate whether the process is to be restarted or not, while updating the process information saved in the save area to the latest state if the process is not to be restarted [col. 10, lines 14-20]; process information save area releasing means for nullifying the saved process information, at termination of the user process [1004]; main memory initialization means for initializing, at restart of an OS, a main memory area of the main memory device used by the OS while not restarting the save area used by the user process for which the restart flag is set not to restart [col. 10, lines 6-10]; and process restoration means for restoring, after the restart of the OS, the saved process information of the user process for which the restart flag is set not to restart in the OS [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

30. As to claim 24, Sekiguchi discloses means for searching, when a restart flag is set for process information relating to a certain user process to designate whether the process is to be restarted or not, all the user processes belonging to the same user application program as the user process, and setting restart flags in the process information relating to all the searched user processes to the same value [col. 8, lines 19-37].

31. As to claim 25, Sekiguchi discloses means for searching all the user processes belonging to the user application program as the user process, when a restart flag is set for process information relating to a certain user process to designate whether the process is to be restarted or not, and setting restart flags in the process information relating to all the searched user processes to the same value [col. 8, lines 19-37].

32. As to claim 26, Sekiguchi discloses means for starting up the OS from an OS main memory image stored in a nonvolatile storage portion forming a part of the main memory device [FIG. 1, "REBOOT LOADER" in main memory device 102].

33. As to claim 27, Sekiguchi discloses means for copying, at every occurrence of a write access from the OS to the nonvolatile storage portion during system operation, data in an address range having a predetermined width including the address at which the write access has occurred from the nonvolatile storage portion to a substitute area ensured in a readable/writable main memory portion forming part of the main memory device, and for converting subsequent accesses to the address range having the predetermined width to accesses to the substitute area [col. 9, line 39—col. 10, line 5].

34. As to claim 28, Sekiguchi discloses a method for restarting an OS in a computer in which a first OS memory area for loading an OS and a process memory area for loading processes are allocated on a main memory, and the OS and the processes are loaded in the respective areas, the

OS restart method comprising: a first step of acquiring process information, that is information for the OS to manage the processes, from the first OS memory area and storing the same in a save area provided in a predetermined storage device [col. 13, lines 14-52, and col. 9, line 39—col. 10, line 5]; a second step of initializing the first OS memory area while holding the process memory area [col. 10, lines 6-10]; a third step of allocating a second OS memory area on the main memory and loading the OS therein [col. 10, lines 6-10]; and a fourth step of updating the process information in the OS memory area according to the process information stored in the first step [col. 10, line 29 – col. 11, line 35]. Kang further teaches the information that is stored comprises the status of programs executing on the computer, which comprises the process conditions [col. 10, lines 6-31].

35. As to claim 29, Sekiguchi discloses selecting a process to be held from the processes loaded in the process memory area [col. 10, lines 56-58]; and initializing the process memory area allocated to the processes not selected [col. 10, lines 56-58].

36. As to claim 30, Sekiguchi discloses the save area is provided on the main memory [102].

37. As to claim 31, Sekiguchi discloses information indicating whether each of the processes is to be restarted or not is stored in the save area together with the process information of the relevant process [FIG. 6].

38. As to claim 32, Sekiguchi discloses information indicating whether each of the processes is to be restarted or not is stored in a separate storage device from the storage device having the save area provided therein [105].

39. As to claim 33, Sekiguchi discloses processing to generate, update and release the save area are executed in accordance with the generation, switching and termination of a process on the storage device having the save area provided therein [col. 11, lines 21-28].

40. As to claim 34, Sekiguchi discloses preliminarily preparing a nonvolatile storage device storing an image of the OS when it is loaded in the main memory, the third step referring to the image stored in the nonvolatile storage device to load the OS in the main memory [111].

41. As to claim 35, Sekiguchi discloses loading a process including a plurality of processes associated with one application program in the process memory area; selecting a process to be held from among the processes loaded in the process memory area; and initializing the process memory area allocated to the processes other than the selected process and the other processes associated with the same application program as the selected process [col. 10, line 29 – col. 11, line 35].

Response to Arguments

42. Applicant's arguments with respect to claims 1-35 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

43. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ERIC CHANG whose telephone number is (571)272-3671. The examiner can normally be reached on M-F 9:00-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Huynh can be reached on (571) 272-4147. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eric Chang/
Examiner, Art Unit 2116

/KIM HUYNH/
Supervisory Patent Examiner, Art Unit 2116